

カルマンフィルタ 実装編

B3 勉強会

2017.11.22

サンプルプログラム配布ページ

• https://github.com/NaohiroTawara/B3_seminor2017/tree/master/Kalman/

• Gitを使えば下記でまとめてダウンロードできます

```
$ git clone https://github.com/NaohiroTawara/B3_seminor
```

• または、前回cloneしたディレクトリ上で以下を実行

```
$ git pull
```

基本課題：魚釣り過程 ローカルレベルモデル

- 魚釣り過程を以下のローカルレベルモデルで定義して、各時刻に釣った魚の数から湖の中にいる魚の数を推定してみよう

1. 観測方程式 $y(k) = x(k) + w(k)$

2. 状態方程式 $x(k) = x(k-1) + v(k-1)$

観測値 $y(k)$ ：時刻 k に釣った魚の数

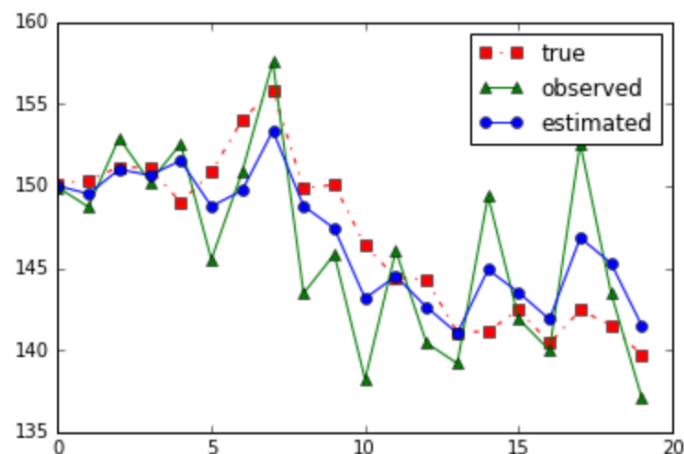
状態 $x(k)$ ：時刻 k に湖の中にいる魚の数

ただし、ノイズ $w(k), v(k)$ は平均0, 分散1の正規分布に従うとする

$$w(k) \sim N(0, 1), \quad v(k) \sim N(0, 1)$$

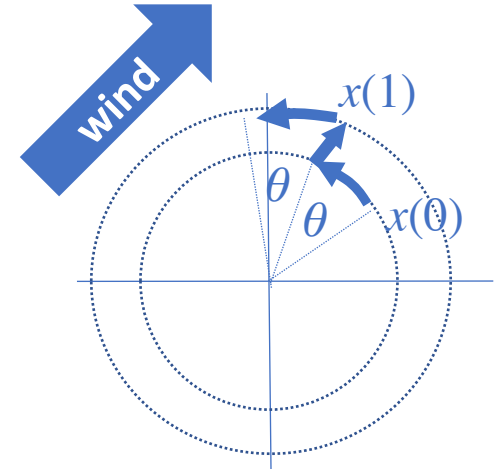
基本課題：魚釣り過程 ローカルレベルモデル 実行例

- 緑が観測された釣った魚数 $y(t)$
- 赤が真の湖の中にいる魚数 $x(t)$
- 青が推定された湖の中にいる魚数 $\hat{x}(t)$



応用課題：ロボットの位置推定

- GPSを搭載したロボットが単位時間あたりに θ [radian/sec] だけ時計回りで等速度で旋回しています
- ロボットは毎時刻GPSから座標値 $\mathbf{x}(t) = (x_1(t), x_2(t))$ を取得できますがこれには平均 0, 分散 1 の正規分布に従う誤差が含まれます
- また、フィールドには常に北東方向に一定の風が吹いておりロボットは移動時に北東方向 $\mathbf{u} = (1, 1)$ を中心に分散 1 程度の分散で流されます
- 初期位置は $(10, 10)$ とします
- 以上の条件において取得されるGPS情報 \mathbf{x}_t を用いてロボットの真の位置 $\mathbf{y}_t = (y_1(t), y_2(t))$ をカルマンフィルタで推定してください



応用問題：ロボットの位置推定 諸式

$$\mathbf{y}(k) = \mathbf{B}\mathbf{x}(k) + \mathbf{C}u(k) + \mathbf{w}(k)$$

$$\mathbf{x}(k) = \mathbf{A}\mathbf{x}(k-1) + \mathbf{v}(k-1)$$

$$\mathbf{x}(k) = \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix}, \quad \mathbf{y}(k) = \begin{pmatrix} y_1(k) \\ y_2(k) \end{pmatrix}$$

$$\mathbf{A} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

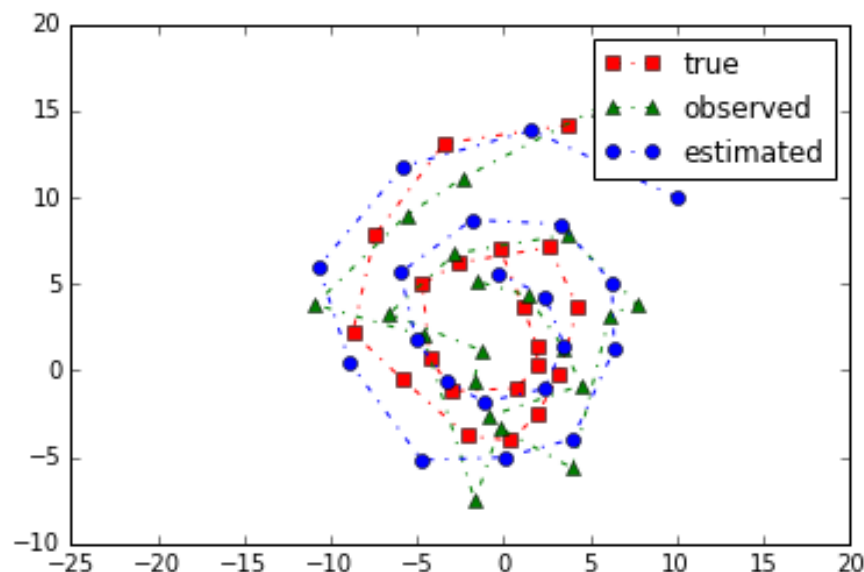
$$\mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{w}(k) \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right),$$

$$\mathbf{v}(k) \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$$

応用課題：ロボットの位置推定 実行例

- 緑が観測されたGPS座標 $y(t)$
- 赤が真の座標 $x(t)$
- 青が推定された座標 $\hat{x}(t)$



報告事項

- 課題 1

- 魚釣り過程のサンプルプログラムを回して実行結果を確認してください

```
$ jupyter notebook Kalman.ipynb
```

- このとき初期状態や状態方程式と観測方程式の分散を変えると結果がどうなるか確認してください

- 課題 2

- 課題 1 を参考にロボットの位置推定を行うためのカルマンフィルタ`lkf_robot()`を完成させてください
- 課題 1 と同様に初期状態や状態方程式と観測方程式の分散を変えると結果がどうなるか確認してください

- 以上をpdfにまとめ, 11/28の23:59までに
<https://waseda.app.box.com/folder/41933809603>
の第二回課題提出用フォルダに「氏名.pdf」でアップロードしてください